(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 727 746 A2

# (12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
**21.08.1996 Bulletin 1996/34**

(51) Int. Cl.[6]: **G06F 12/14**

(21) Application number: **96102115.1**

(22) Date of filing: **13.02.1996**

(84) Designated Contracting States:
**DE FR GB**

(30) Priority: **14.02.1995 JP 25707/95**

(71) Applicant: **FUJITSU LIMITED**
**Kawasaki-shi, Kanagawa 211 (JP)**

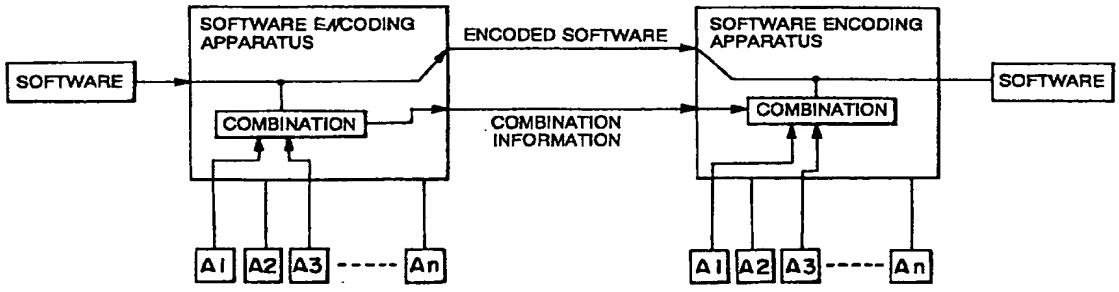
(72) Inventors:
- **Akiyama, Ryota,**
  **c/o Fujitsu Limited**
  **Kawasaki-shi, Kanagawa, 211 (JP)**
- **Yoshioka, Makoto,**
  **c/o Fujitsu Limited**
  **Kawasaki-shi, Kanagawa, 211 (JP)**

(74) Representative: **Schulz, Rütger, Dr. Dipl.-Phys.**
**Patentanwälte Mitscherlich & Partner,**
**Sonnenstrasse 33**
**80331 München (DE)**

## (54) Method and system for encoding and decoding software

(57) When encoding software, two or more encoding algorithms are employed. Conversely, for software decoding processing, decoding algorithms are prepared that correspond to these two or more encoding algorithms. Then, in software encoding processing, encoded algorithm combination information is transferred to the software decoding processing along with the encoded software. During software decoding processing, opposite algorithms possessed by the decoding means are selected based on the previously mentioned algorithm combination information, and the previously mentioned encoded software is decoded.

FIG. 1






EP 0 727 746 A2

## Description

### BACKGROUND OF THE INVENTION

The present invention relates to techniques for the encoding of character, voice, animation, still image, program, or other data, and to techniques for the decoding of these kinds of encoded data.

As for software distribution formats, the method of encrypting character, voice, animation, still image, program or other data and storing this data on floppy disks, CD-ROMs, magneto-optical disks (MO), or other media, and selling these media, may be employed, or the method of encoding said data and delivering this data to users through a communications circuit, as well as other methods, may be employed.

In previous encoding methods of this kind, the provider has, for example, encoded a program with a single algorithm and provided it to the user, and the user has performed the decoding operation using so-called "key" program provided from said user.

However, when key program possessed by the user is decoded by a wrongful user, there is a fear that a copy tool will be created based on the results of this decoding. If a large amount of these copy tools are circulated, the provider will be forced to incur a substantial loss.

The present invention has been created bearing this problem in mind, by noting that this type of software has the special characteristic of undergoing upgrades after only a brief period of time. The present invention therefore provides an encoding combination method whereby the encoding algorithm may be changed periodically or at every software upgrade.

### SUMMARY OF THE INVENTION

The present invention employs two or more encoding basic algorithms for encoding software. Conversely, for software decoding processing (the software decoding apparatus), decoding basic algorithms are prepared that correspond to these two or more encoding algorithms. Then, the software encoding apparatus transfers along with the encoded software combination information concerning the encoded algorithms to the software decoding means (the software decoding apparatus).

The above-mentioned encoding basic algorithm means a minimum processing necessary to convert input data row into random data row. For example, convert input data row into rearrangement data row or replacement data row. In addition, exclusive-OR of input data row and other random data row can be output.

Based on the previously mentioned algorithm combination information, the software decoding means selects decoding basic algorithms that it possesses, and decodes the previously mentioned encoded software.

Within the encoding means, when the software is inputted into the software encoding apparatus, it is encoded with a voluntarily selected basic algorithm combination (for example, A1 and A3). Software that has been encoded in this way is distributed to the user in the form of a CD-ROM or via a communications circuit. The user decodes this encoded software with a decoding apparatus that he or she possesses. At this time, the user decodes the relevant encoded software based on the encoding combination information (for example, A1 ¦ ¦ A3) used by the encoding apparatus. This combination information may be delivered to the user on the same medium as the encoded software, or on a different medium, or it may be communicated to the user along with non-illustrated key information (K).

Moreover, the combination information may be encoded by using an algorithm before updating.

In this way, if, for example, each of the individual algorithms are relatively easy to analyze, the combining of these algorithms increases the difficulty of analysis. Therefore, if, for example, each individual algorithm may easily be grasped on its own, since there is an enormous amount of possible algorithm combinations, it would require significant time and effort to analyze a combination of algorithms, making such an analysis difficult in reality. Even if there were to be a possibility of such an analysis occurring, since for every software upgrade, the software is encoded with a new combination of algorithms, the recent software upgrade cycle, whereby software is upgraded after a relatively short period of time, is sufficient to cover for such a possibility.

In fact, a lot of hackers should challenge the combination of the encoded algorithm of infinity number and they cannot help abandoning the decipherment work of software.

### BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a theoretical diagram of the present invention.

Fig. 2 is a configuration diagram of the hardware of a preferred embodiment of the present invention.

Fig. 3 is a functional block diagram of the system configuration of a preferred embodiment of the present invention.

Fig. 4 is an explanatory diagram showing an algorithm table of a preferred embodiment of the present invention.

Fig. 5 is an explanatory diagram showing a concrete example of a transposition type algorithm.

Fig. 6 is an explanatory diagram showing a concrete example of a character conversion type algorithm.

Fig. 7 is an explanatory diagram showing a concrete example of an exclusive algorithm.

Fig. 8 is an explanatory diagram showing a concrete example of a multiplication type algorithm.

Fig. 9 is an explanatory diagram showing a concrete example of DES type combination processing.

Fig. 10 is an explanatory diagram showing a concrete example of an ENIGMA combination.

Fig. 11 is an explanatory diagram showing a concrete example of a case where a transposition type

algorithm (A1) is combined with a character conversion type algorithm (A3) for the performing of ENIGMA type combination processing.

Fig. 12 is an explanatory diagram showing a concrete example of a case where encoded software that has been encoded with the combination processing of Fig. 11 is decoded.

Fig. 13 is an explanatory diagram showing the hardware configuration of this invention.

## DESCRIPTION OF THE PREFERRED EMBODIMENT

Prior to an explanation of a detailed preferred embodiment of the present invention, the basic theory behind the present invention will be explained.

As shown in Fig. 1, in the present invention, when software is to be encoded, two or more encoding basic algorithms are employed. Conversely, for software decoding processing (the software decoding apparatus), decoding basic algorithms are prepared that correspond to these two or more encoding basic algorithms. Furthermore, the software encoding apparatus transfers along with the encoded software combination information concerning the encoded basic algorithms to the software decoding means (the software decoding apparatus).

Based on the previously mentioned basic algorithm combination information, the software decoding means selects decoding basic algorithms that it possesses, and decodes the previously mentioned encoded software.

In this way, when the software is inputted into the software encoding apparatus, it is encoded with a voluntarily selected basic algorithm combination (for example, A1 and A3). Software that has been encoded in this way is distributed to the user in the form of a CD-ROM or via a communications circuit. The user decodes this encoded software with a decoding apparatus that he or she possesses. At this time, the user decodes the relevant encoded software based on the encoding combination information (for example, A1 ¦ ¦ A3) used by the encoding apparatus. This combination information may be delivered to the user on the same medium as the encoded software, or on a different medium, or it may be communicated to the user along with non-illustrated key information (K).

Fig. 2 and Fig. 13 show the hardware configuration of the software encoding apparatus that is a preferred embodiment of the present invention.

The IC card slot (ICR), the floppy disc drive (FD) and hard disk drive (not shown in figures ) are installed in the main body of the computer (BDY).

The modem (MDM) and the receiver (not shown in figures) can be built into the main body of the computer (BDY) and the data be transmitted or received through cable or the wireless route through the ANT.

The keyboard (KEY) and the mouse (MOU) are connected with the main body of the computer (BDY) as an input device.

The display (CRT) and the printer (PRN) are connected with the main body of the computer (BDY) as an output device.

The optical disk drive (MOD) is connected with the main body of the computer (BDY) further as an external storage device.

Within the diagram, BUS is the bus, indicating both the control bus and the data bus. CPU is the central processing unit, and a 32-bit or a 64-bit processor is used. MEM is the memory, which is set by the encoding combination table, to be described below, and an address specified by the operation domain. KEY is the keyboard, which is used to perform the input of commands into this apparatus, and to input key information (K). FD is the floppy disk apparatus, which reads out the software for performing encoding processing.

Although only a floppy disk is shown as the software provision means, such means is not limited to this alone: a magneto-optical disk or a communications circuit may also be used. Moreover, the memory MEM or a buffer within the central processing unit CPU may also function as the software provision means.

ICR is the IC card reader, and it is possible to equip the system with a PCMCIA card that conforms to the JEIDA (Japan Electronic Industry Development Association) standard. In the present preferred embodiment, the algorithms A1, A2, A3 . . . An to be discussed below are provided by being stored on this IC card. It is desirable to perform a physical protection for this IC card; for example, the algorithm data may be completely erased by a wrongful opening of the card by a user, or card read-out may be prevented. Furthermore, the configuration of the multiple algorithms on this card may be changed following a prescribed fixed period. If the arrangement or contents of the algorithms stored in this card are renewed periodically, the decoding of the software may be limited to a certain period of time, enabling it to be used as demonstration software, or as specific-system software for limited uses.

MOD is the magneto-optical disk apparatus, which is able to read from and write onto a magneto-optical disk with a capacity of 100 megabytes or more. Also, IF is the communication interface, which is connected to an external communications circuit.

Besides the above apparatus configuration, a CD-ROM sampling apparatus, for example, may also be connected.

Although the above is an explanation of the software encoding apparatus of Fig. 2, the software decoding apparatus has the same configuration. However, when this apparatus is used as the software decoding apparatus, a CD-ROM drive may also be provided. In other words, as a medium for distributing voice, still image, animation, and other large amounts of data, the CD-ROM is appropriate.

Fig. 3 shows a functional block diagram of the system configuration of the present preferred embodiment. Within the diagram, the upper tier shows the functions of the software encoding apparatus, while the lower tier

shows the functions of the software decoding apparatus.

To explain the relationship between the functions shown in Fig. 3 and the hardware explained with Fig. 2, the algorithm program (A1, A2 . . . An) shown in Fig. 3 is provided by being stored on the IC card. Also, selection processing, combination processing, and combination program processing are functions realized by using the central processing unit (CPU). Key information (K) is inputted using the keyboard (KEY). Also, the software storage file is inputted via the floppy disk apparatus (FD) or the magneto-optical disk apparatus (MOD), and is written into the memory (MEM).

Next, Fig. 3 will be used to explain software encoding processing.

First, the basic algorithms to be used for software encoding processing are determined through selection processing by the central processing unit (CPU). Here, processing is performed whereby two or more algorithm programs are determined upon. The determination of algorithms may be performed, for example, by using algorithm tables, such as those shown in Fig. 4, which have been established within the memory (MEM).

As shown in the same diagram, n number (A1, A2 . . . An) of algorithms are provided in matrix form within the algorithm table. In this way, when n number of basic elements (algorithms) are used to form separate combinations of two algorithms, when overlapping is permitted, $n^2$ algorithm combinations can be formed.

Conversely, in the case where any voluntary number of n number of basic elements (algorithms) are combined in tandem connections, when different line combinations are performed for each element (algorithm), a maximum of n! number of algorithm combinations may be obtained. Furthermore, when overlapping line combinations (exchanges) are permitted, a maximum $n^n$ algorithm combinations may be formed.

For example, algorithms A1 and A2 may be combined to form A1 || A2, or a combination of three algorithms may be performed to form, for example, A1 || A2 || A3. Also, the same algorithm may be combined in L steps, such as A1 || A1 || . . . || A1.

In this way, with the present preferred embodiment, since a multiple number of algorithm combinations may be formed, even if, for example, individual basic algorithms are easily analyzable on their own, by combining these basic algorithms together, an algorithm that is difficult to analyze may be used.

Next, a concrete algorithm example will be explained. Fig. 5 shows a concrete example of a transposition type basic algorithm. In other words, with this algorithm, a key linkage switching switch is established as a program, and for every 8 bits of data, bit locations are transposed and then outputted. Key information (K) provided from an external source controls the transposition locations. Moreover, although input/output is shown as an 8-bit configuration in the diagram, this is not limited to such a configuration. This kind of transposition type algorithm may, for example, be registered as "A1" in the algorithm table described previously.

Fig. 6 shows a concrete example of a character conversion type basic algorithm. With this algorithm, output data corresponding to input data is placed in a table and preserved. Moreover, although input/output is shown as a 3-bit configuration in the diagram, naturally, this is not limited to such a configuration. This kind of character conversion type algorithm may, for example, be registered as "A2" in the algorithm table described previously.

Fig. 7 shows a concrete example of an exclusion algorithm. With this algorithm, for each bit of data, for example, key information (K) may be used to perform exclusive logic processing. Moreover, although input/output is shown as a 3-bit configuration in the diagram, naturally, this is not limited to such a configuration. This kind of exclusion algorithm may, for example, be registered as "A3" in the algorithm table described previously.

Fig. 8 shows a concrete example of a multiplication type algorithm. In this diagram, when 8-bit data is inputted, after this inputted data is multiplied with the multiplier device using the key information (K), data that has been masked with the output masking circuit is outputted. This kind of multiplication type algorithm may, for example, be registered as "A4" in the algorithm table described previously.

When algorithms from the multiple algorithms as those described above are selected with the "selection processing" of the central processing unit (CPU), combination processing (encoding execution means) is performed. Fig. 9 and Fig. 10 show concrete examples of this combination processing.

Fig. 9 shows an example of a DES type combination.

Within the diagram, when software data is inputted, it is divided into groups of a prescribed number of bits (for example, after every 8 bits), and processing is performed on these groups. Here, the said 8-bit data is divided into left-half data (D1) which comprises the 4 high-order bits, and right-half data (D2) which comprises the 4 low-order bits.

Then, the right-half data (D2) is processed with the A1 algorithm, and is outputted as encoded data C1.

The left-half data (D1) undergoes exclusive logic processing along with the previously mentioned encoded data (C1), and is outputted as encoded data (D1+C1).

Conversely, the exclusive logic processing output data that includes the left-half data (D1) and the A1 algorithm output data (C1) is processed with the A2 algorithm, and is output as encoded data C2. This encoded data C2 undergoes exclusive logic processing along with the right-half data (d2), and is outputted as encoded data (D2+C2).

Although the decoding processing for Fig. 9 is not shown in a diagram, it is sufficient to prepare a system whereby an A1 decoding algorithm replaces the A1 pro-

gram, and an A2 decoding algorithm replaces the A2 program shown in Fig. 9.

Fig. 10 shows an example of an ENIGMA combination. In this example, after inputted software is primarily changed with the A1 algorithm, it undergoes a secondary change with the A2 program.

Also, as shown in the right half of the diagram, in the case where this is to be decoded, it is first primarily decoded with the decode algorithm $A2^{-1}$ of algorithm A2. By then performing a secondary decoding with the decode algorithm $A1^{-1}$ of algorithm A1, the software can be returned to its original form.

Next, by using Fig. 11, an even more concrete preferred embodiment of an algorithm is explained. In this case, a transposition type algorithm (A1) is combined with a character conversion type algorithm (A2), and ENIGMA type combination processing is performed.

First, key information (K) is provided to the key linkage switching switch. This key information (K) is information that prescribes how each bit of inputted 8-bit data will be transposed for output. In the diagram, the following key information (K) settings are provided to the key linkage switching switch: #1 bit → #7 bit, #2 bit → #3 bit, #3 bit → #2 bit, #4 bit → #5 bit, #5 bit → #1 bit, #6 bit → #8 bit, #7 bit → #6 bit, #8 bit → #4 bit.

Here, when software data (ordinary character data) "0Fh"--in other words, "00001111"--is provided, with the key linkage switching switch as prescribed with the previously described key information (K), the following transpositions occur: the #1 bit "0" becomes the #7 bit, the #2 bit "0" becomes the #3 bit, the #3 bit "0" becomes the #2 bit, the #4 bit "0" becomes the #5 bit, the #5 bit "1" becomes the #1 bit, the #6 bit "1" becomes the #8 bit, the #7 bit "1" becomes the #6 bit, and the #8 bit "1" becomes the #4 bit. As a result, the switching result with algorithm A1 is "10010101", in other words, "95h".

Next, the primary encoded data "95h" that has been encoded with algorithm A1 then undergoes secondary encoding with the character conversion algorithm A2.

Here, the input data (95h) is first divided into left-half data, which comprises the 4 high-order bits, and right-half data, which comprises the 4 low-order bits. Then, conversion is performed based on each conversion table. As a result, the secondary encoded output data becomes "00100000", in other words, "20h".

Next, as shown in Fig. 3, the above selection processing, combination processing, and combination program processing is performed, and encoded software is created. This encoded software is then saved on a medium such as an MO, a CD-ROM, or a floppy disk, etc., and is sent (delivered) to the user by the provider. Also, it may be transmitted to the user through an interface (I/O) via a communications circuit. Also, at this time, the algorithm selection order code--for example, the previously mentioned code "A1 ¦¦ A2"--is stored on the MO, the CD-ROM, the floppy disk, etc., that is the medium along with the encoded software. Also, in the case of communication transmission, this code may also be sent with the encoded software along the communications route. Furthermore, this selection order code may be communicated from the provider to the user along a separate delivery route than the previously mentioned encoded software; for example, it may be explained verbally over the telephone, etc. Moreover, the selection order code may be provided to the user together with the key information (K).

After the user has received the previously mentioned encoded software and selection order code, with the decoding apparatus on the user side, based on the previously mentioned selection order code information, decoding algorithms are selected, combination processing and combination program processing is performed, and decoded software is obtained.

Fig. 12 shows a concrete example of this decoding processing.

Within the diagram, for every 8 bits (20h) of encoded data from the encoded software, after dividing the 4 high-order bits into left half data, and the 4 low-order bits into the right half data, first character conversion is executed on these with algorithm $A2^{-1}$. The primary encoded data (95h) obtained with this character conversion further undergoes transposition conversion with the transposition type basic algorithm $A1^{-1}$, and decode data "00001111"-in other words, "0Fh"-is obtained.

## Claims

1. A software encoding and decoding method comprising:

   software encoding step for encoding a software by combining at least two encoding basic algorithms; and

   software decoding step for inputting encoded software and for decoding said software with decoding basic algorithms corresponding to said at least two encoding algorithms used in said encoding step;

   wherein said software encoding processing step includes a step of transferring information concerning said combination of said at least two algorithms used for encoding together with the encoded software; and

   wherein said software decoding processing step includes a step of selecting decoding algorithms based on said information of the combination of said encoding algorithms.

2. The software encoding and decoding method as claimed in claim 1, wherein said software decoding step further includes a step for decoding said encoded software with the information of said combination of the algorithms and key information of the algorithms.

3. A software encoding system comprising:

   software provision means for providing software;

basic algorithm provision means for providing a plurality of algorithms;

selection means for selecting at least two basic algorithms from the plurality of algorithms provided by said algorithm provision means;

encoding execution means for encoding software read out from said software provision means by using said at least two basic algorithms selectively combined with said selection means; and

output means for outputting encoded software outputted from the encoding execution means.

4. The software encoding system as claimed in claim 3, further comprising:

combination sequence recording means for recording the combination sequence of said at least two basic algorithms selected with said selection means; and

wherein said output means outputs a combination sequence data obtained from said combination sequence recording means along with the encoded software.

5. The software encoding apparatus as claimed in claim 3, wherein said encoding execution means divides a read-out software into bit groups of a predetermined number, each bit group having specified number of bits, performs encoding processing in a parallel fashion for each of said bit group of bits with said at least two basic algorithms that is selectively combined with said selection means, and combines said bit groups that has been combined.

6. The software encoding apparatus as claimed in claim 3, wherein said encoding execution means divides a read-out software into bit groups of a predetermined number, each bit group having specified number of bits, performs encoding processing for each of said bit group in order with said at least two basic algorithms that is selectively combined with said selection means, and combines said bit groups that has been combined.

7. A software decoding system comprising:

encoded software provision means for providing software that is encoded;

algorithm provision means for providing a plurality of basic algorithms;

selection means for selecting at least two basic algorithms that are necessary for decoding of software from the plurality of basic algorithms provided by said algorithm provision means;

combination sequence recording means for recording the combination sequence of said at least two algorithms that is selected with said selection means; and

decoding execution means for decoding encoded software provided from said encoded software provision means by using said at least two algorithms that is selectively combined with said selection means.

8. The software decoding means as claimed in claim 7, wherein said algorithm provision means is an execution program of said at least two algorithms and is maintained within a recording medium that is physically protected from an external source.

FIG. 1

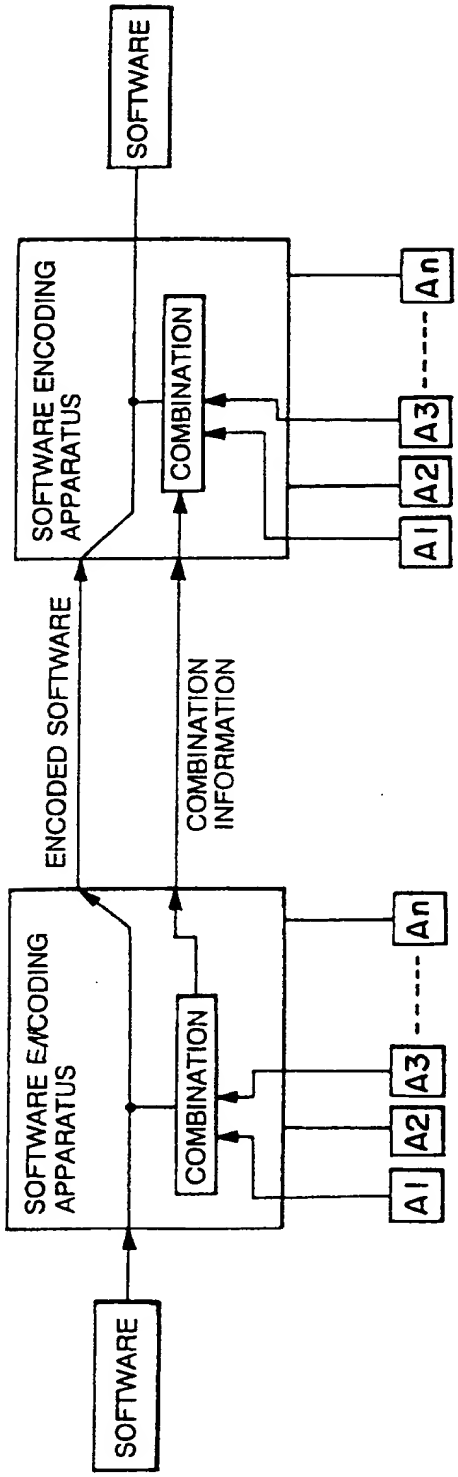

SOFTWARE
SOFTWARE ENCODING APPARATUS
COMBINATION
A1
A2
A3
An
ENCODED SOFTWARE
COMBINATION INFORMATION
SOFTWARE ENCODING APPARATUS
COMBINATION
A1
A2
A3
An
SOFTWARE

FIG. 2

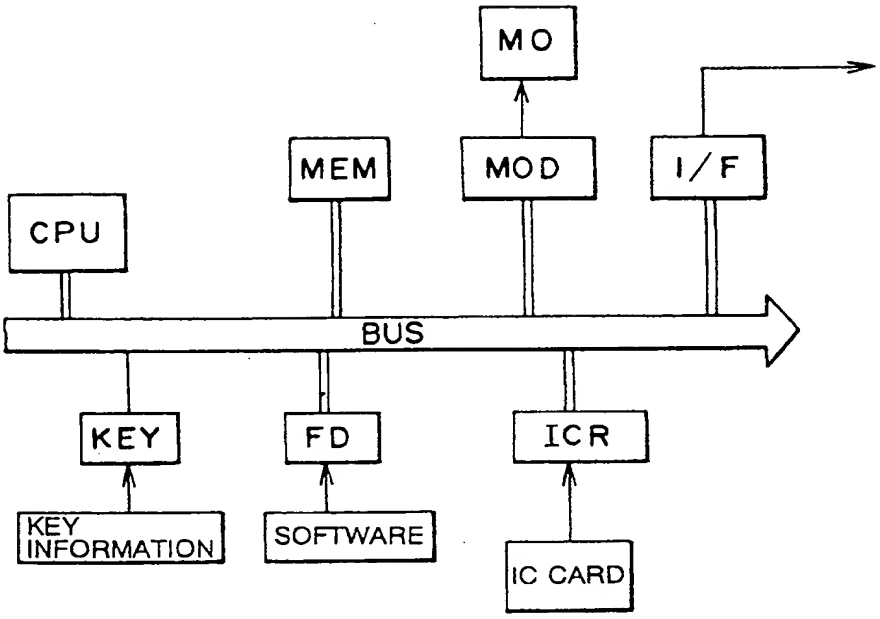

MO
MEM
MOD
I/F
CPU
BUS
KEY
FD
ICR
KEY INFORMATION
SOFTWARE
IC CARD

FIG. 3

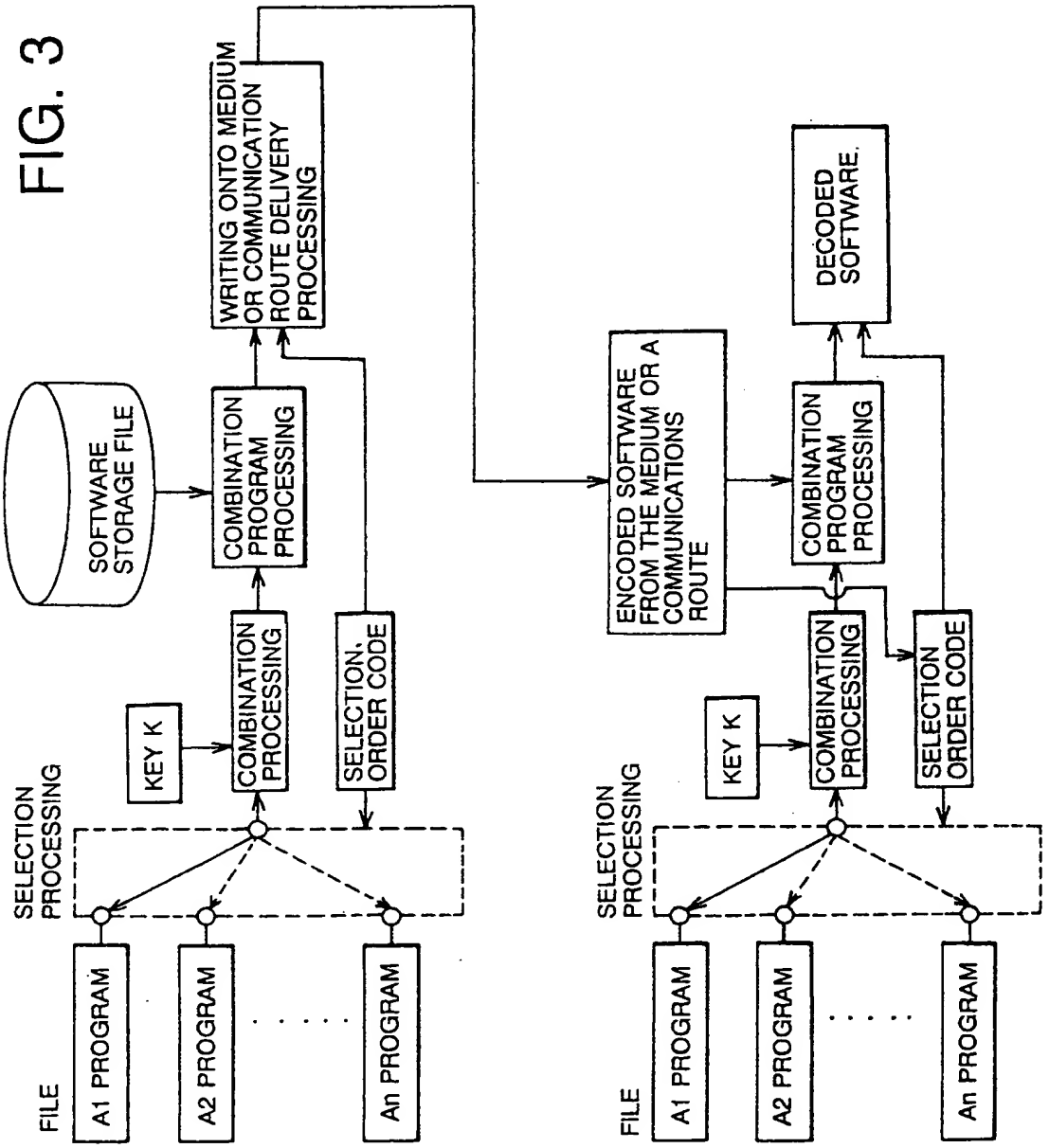

SOFTWARE STORAGE FILE
COMBINATION PROGRAM PROCESSING
WRITING ONTO MEDIUM OR COMMUNICATION ROUTE DELIVERY PROCESSING
KEY K
COMBINATION PROCESSING
SELECTION ORDER CODE
SELECTION PROCESSING
FILE
A1 PROGRAM
A2 PROGRAM
An PROGRAM
ENCODED SOFTWARE FROM THE MEDIUM OR A COMMUNICATIONS ROUTE
DECODED SOFTWARE

FIG. 4

| | | DIFFERENT TYPES OF ALGORITHMS | | | |
|---|---|---|---|---|---|
| | | A 1 | A 2 | · · · · · · | An |
| DIFFERENT TYPES OF ALGORITHMS | A 1 | A11 | A12 | | A1n |
| | A 2 | A21 | A22 | | A2n |
| | · · · · · · | | | | |
| | An | An1 | An2 | | Ann |

FIG. 5

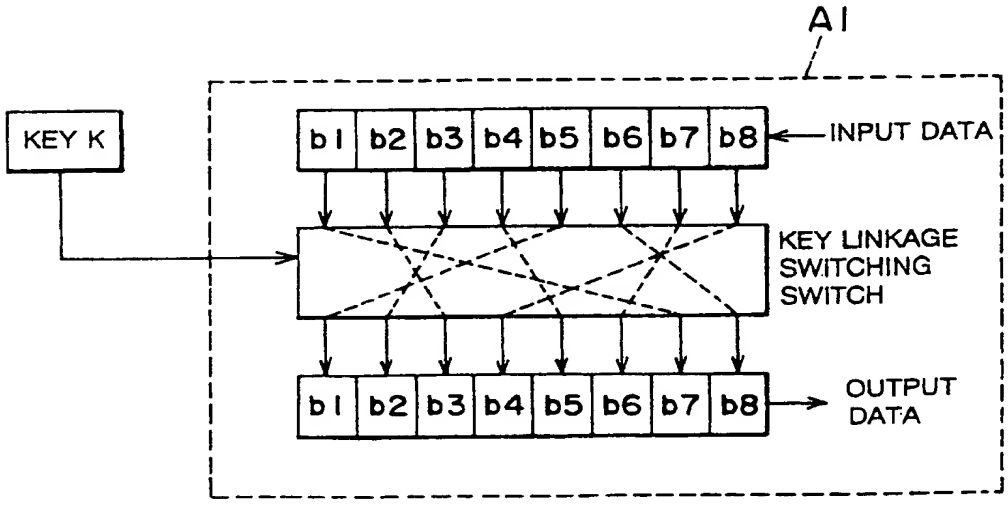


FIG. 6

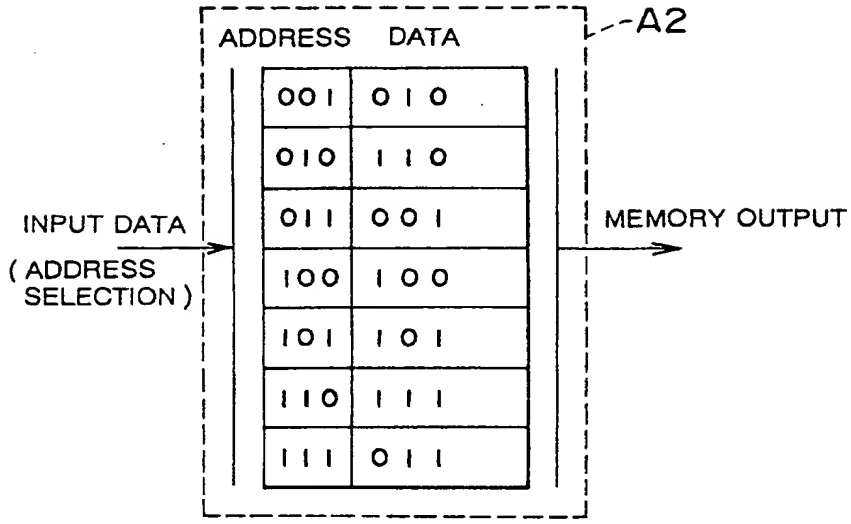

A2
ADDRESS
DATA
001
010
010
110
011
001
100
100
101
101
110
111
111
011
INPUT DATA
( ADDRESS SELECTION )
MEMORY OUTPUT

FIG. 7

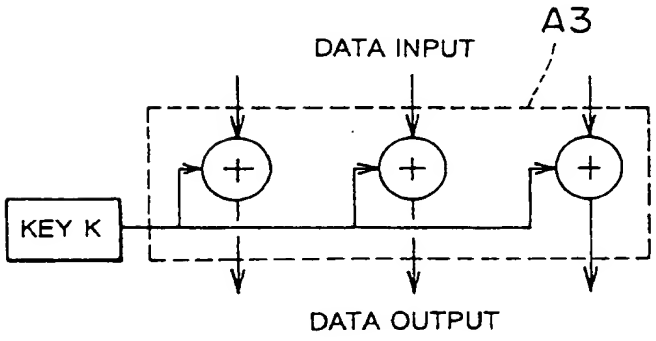

A3
DATA INPUT
KEY K
DATA OUTPUT

FIG. 8

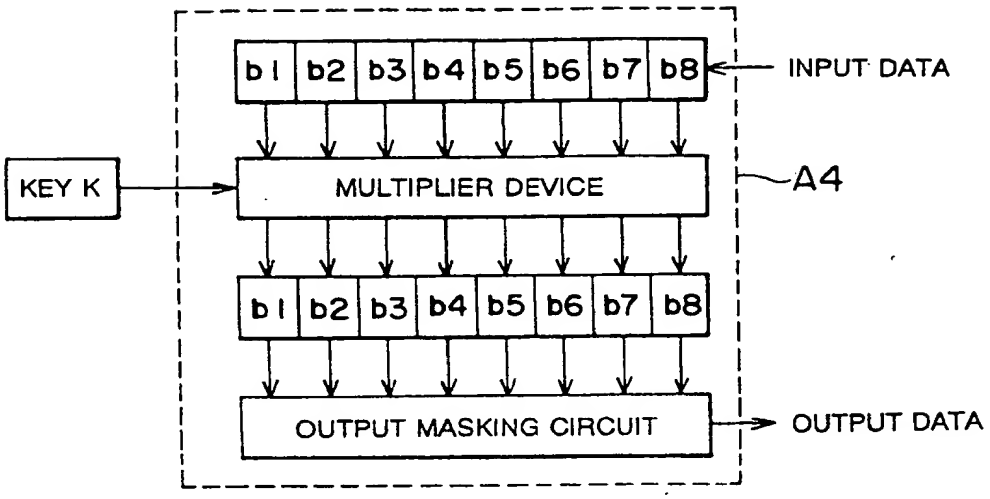

b1 b2 b3 b4 b5 b6 b7 b8
INPUT DATA
KEY K
MULTIPLIER DEVICE
A4
b1 b2 b3 b4 b5 b6 b7 b8
OUTPUT MASKING CIRCUIT
OUTPUT DATA

FIG. 9

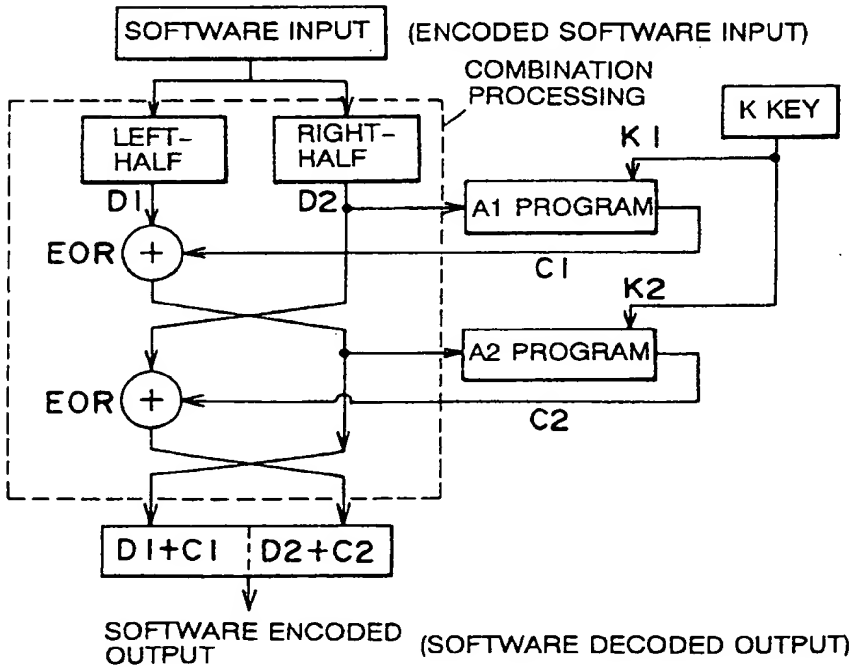

SOFTWARE INPUT
(ENCODED SOFTWARE INPUT)
COMBINATION PROCESSING
K KEY
LEFT-HALF
RIGHT-HALF
K1
D1
D2
A1 PROGRAM
EOR
C1
K2
A2 PROGRAM
EOR
C2
D1+C1
D2+C2
SOFTWARE ENCODED OUTPUT
(SOFTWARE DECODED OUTPUT)

FIG. 10

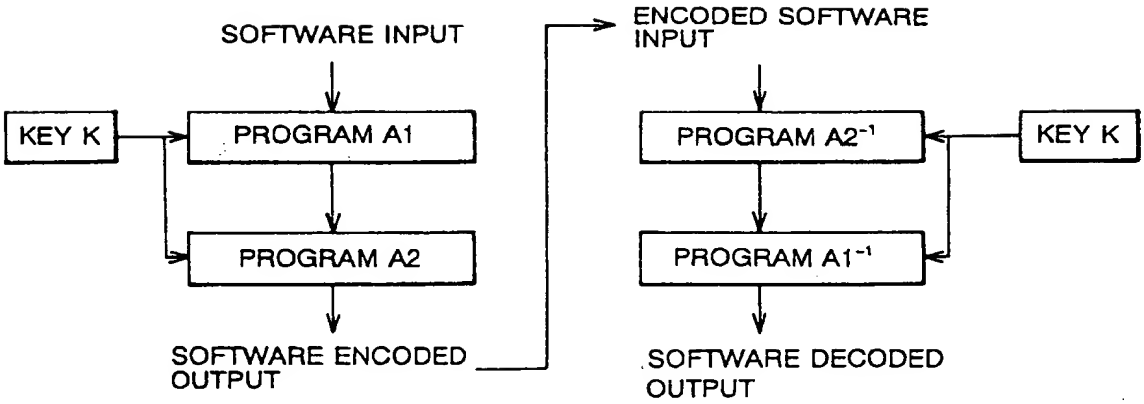

SOFTWARE INPUT
KEY K
PROGRAM A1
PROGRAM A2
SOFTWARE ENCODED OUTPUT
ENCODED SOFTWARE INPUT
PROGRAM A2⁻¹
KEY K
PROGRAM A1⁻¹
SOFTWARE DECODED OUTPUT

FIG. 11

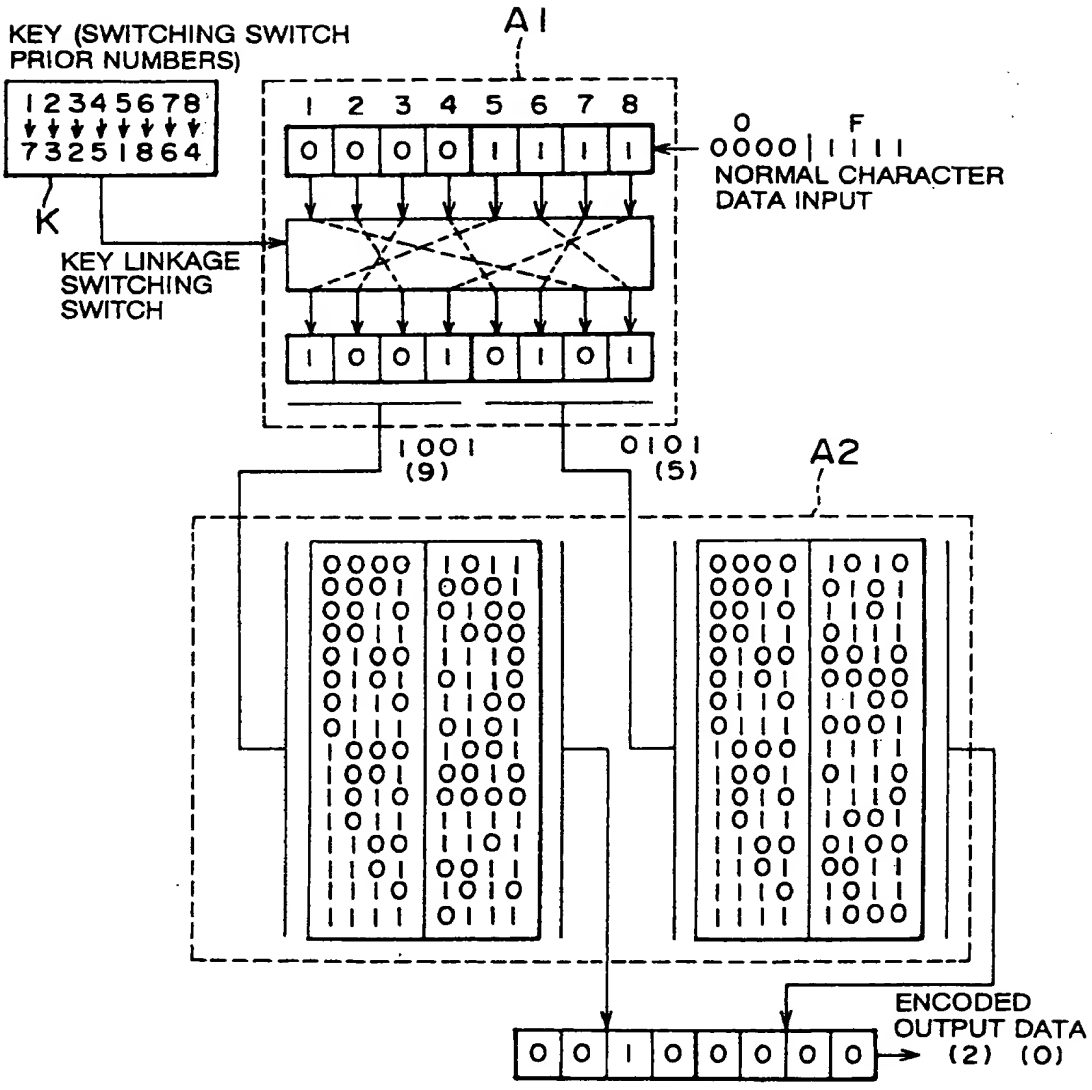

KEY (SWITCHING SWITCH PRIOR NUMBERS)
1 2 3 4 5 6 7 8
7 3 2 5 1 8 6 4
K
KEY LINKAGE SWITCHING SWITCH
A1
1 2 3 4 5 6 7 8
0 0 0 0 1 1 1 1
1 0 0 1 0 1 0 1
0 F
0000 1111
NORMAL CHARACTER DATA INPUT
1001 (9)
0101 (5)
A2
0 0 1 0 0 0 0 0
ENCODED OUTPUT DATA
(2) (0)

FIG. 12

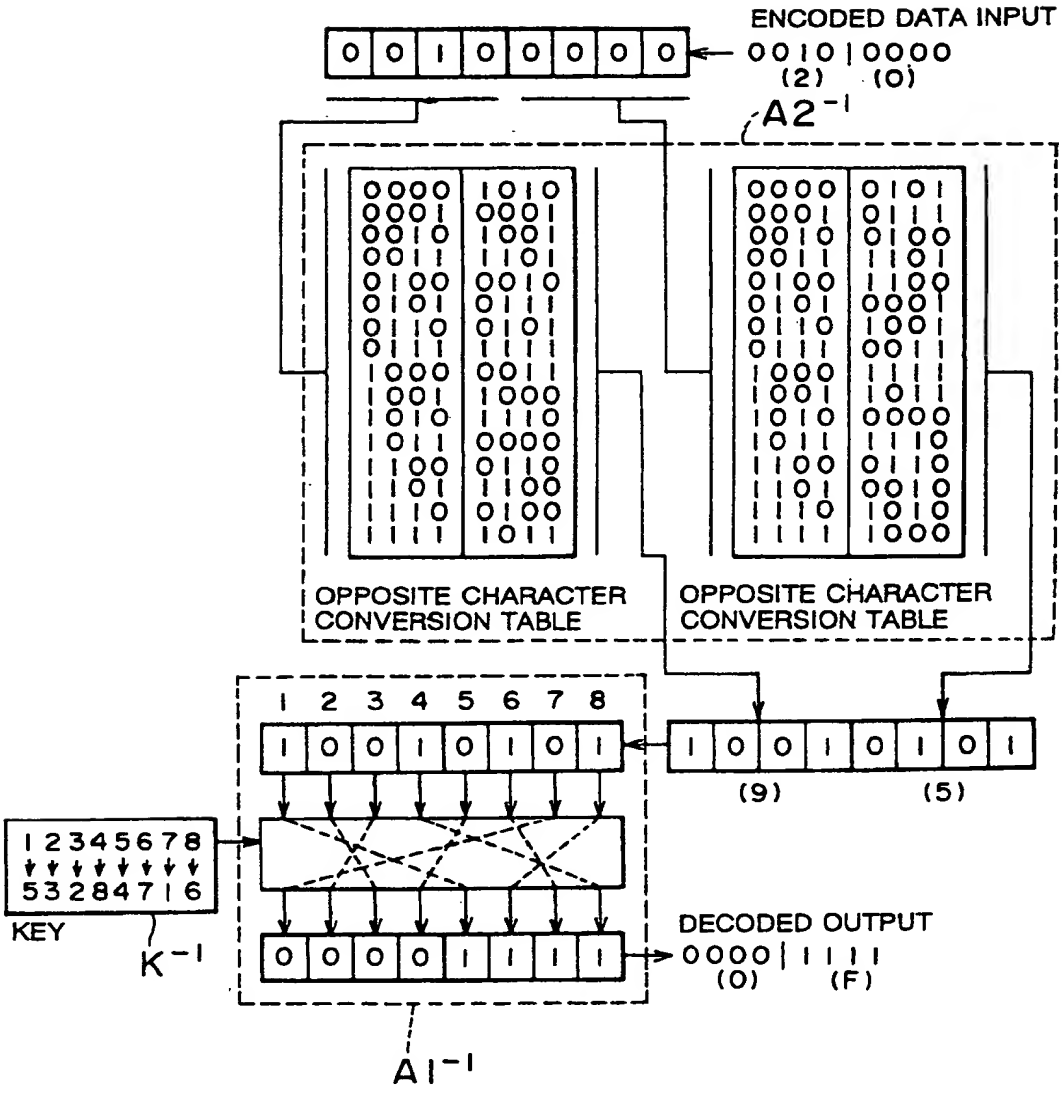
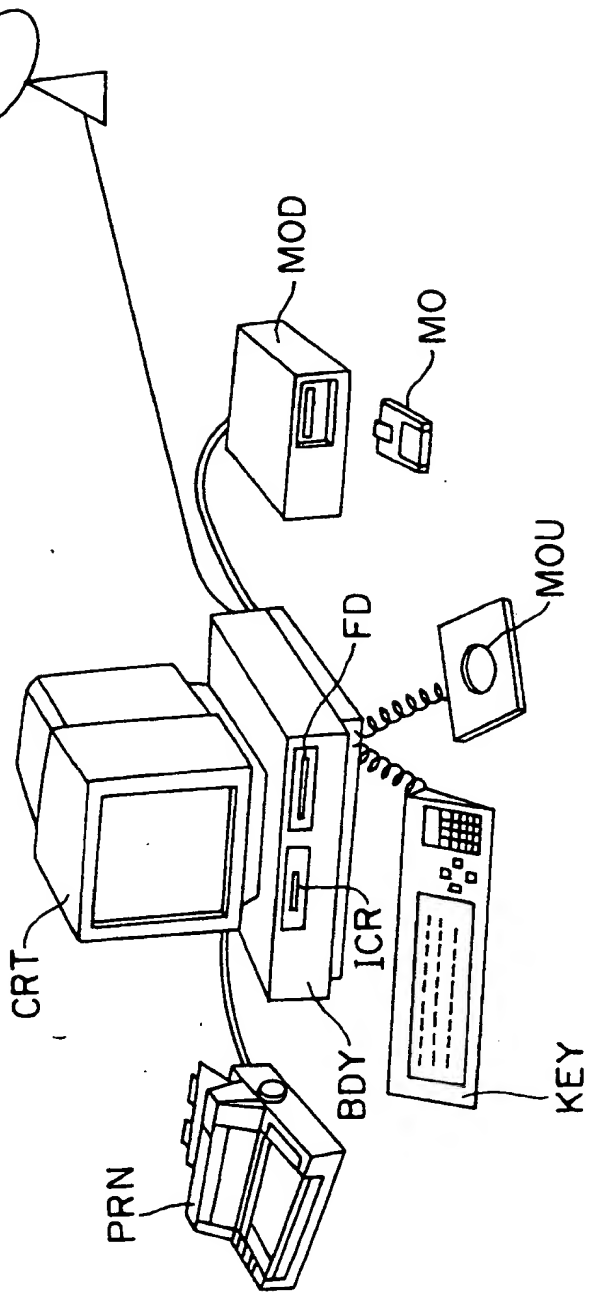

ENCODED DATA INPUT
0 0 1 0 0 0 0 0
0 0 1 0 | 0 0 0 0
(2) (0)
A2⁻¹
OPPOSITE CHARACTER CONVERSION TABLE
OPPOSITE CHARACTER CONVERSION TABLE
1 2 3 4 5 6 7 8
1 0 0 1 0 1 0 1
1 0 0 1 0 1 0 1
(9) (5)
1 2 3 4 5 6 7 8
5 3 2 8 4 7 1 6
KEY
K⁻¹
0 0 0 0 1 1 1 1
DECODED OUTPUT
0 0 0 0 | 1 1 1 1
(0) (F)
A1⁻¹

FIG. 13



# BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

**IMAGES ARE BEST AVAILABLE COPY.**
**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**